Viterbi Algorithm for Noise Line Following Robots

Przemysław Mazurek

Department of Signal Processing and Multimedia Engineering,
West–Pomeranian University of Technology, Szczecin,
26. Kwietnia 10 St., 71126 Szczecin, Poland
przemyslaw.mazurek@zut.edu.pl

Abstract. Image processing and tracking of noise line is considered in this paper. Such line could be obtained for selected application where the line is not direct, but obtained from the image content. The estimation of line allows control of line following robot. Local 2D filter based on standard deviation estimator is applied for the preprocessing of the image. The Viterbi algorithm is applied for the line tracking using assumed Markov model of line. Monte Carlo approach is used for the estimation of the tracking system performance.

1 Introduction

Tracking system are applied in numerous applications and most of them assume Detection and Tracking scheme [3]. The input image is thresholded using fixed or adaptive threshold level and the binary image is obtained. The tracking algorithm adds robustness to estimated trajectory (line) under noisy conditions. There are numerous of tracking algorithms and the Kalman filter is typically applied.

Low quality images are difficult to process due to high noise, because optimal threshold level cannot be determined. Alternative tracking scheme should be applied – TBD (Track–Before–Detect). There are a few groups of TBD algorithms [3,18] and the Viterbi algorithm is assumed [20] in this paper. The tracking is applied for the estimation of cumulative value for trajectory without prior thresholding. TBD algorithms allow raw signal processing, that gives superior performance in comparison with Detection and Tracking scheme based tracking systems.

Specific input image case is related to the images with noise only. It means that the background object trajectory is noise only. Additional preprocessing is necessary, because TBD algorithms assume positive signal of the object, and zero mean of the background noise.

1.1 Related Works

Line following robots are well known and applied in manufactures [5], especially. Typical line following robot application assumes high contrast between the line

R.S. Choraś (ed.), *Image Processing & Communications Challenges 6*,
Advances in Intelligent Systems and Computing 313, DOI: 10.1007/978-3-319-10662-5_14

and the background. Such assumption allows the application of simple optical systems and simple algorithms [7]. The line could be deteriorated and lighting conditions could be variable. Locally variable lighting conditions, and noises related to the deteriorated background and real line, as well as false lines, are the sources of low quality of acquired image. Positive signal of line is assumed and dark background in most cases, but there are applications in which the line is not observed directly. The line could be an edge between two areas or very low quality line related to the real environment. Such cases are typical for the applications [16] like harvesting, trash compacting, snow or sand plowing on airport runways and many more. Road lane estimation is the similar task for automatic control of vehicle or as Lane Departure Warning system. Image processing and pattern recognition techniques are necessary for such local navigation, because GPS systems have low precision for spatial location (about a few meters) for moving platforms

Hough Transform approach for highly deteriorated line is considered in [19] for the road lane estimation as well as in agricultural application [1]. The application of LIDAR for the acquisition of 3D structures for line estimation is considered in [21]. Monte Carlo approach for line estimation is proposed in [15].

The application of the preprocessing using local standard deviation is proposed in [10] and the application of maximal autocovariance is described in [14] for different kind of TBD algorithm – Spatio–Temporal TBD [9].

1.2 Content and Contribution of the Paper

In this paper the Viterbi algorithm is applied for the noise line tracking and the performance of such system is analyzed using Monte Carlo approach, that allows testing many uncorrelated scenarios. The Viterbi algorithm is introduced briefly in Section 2. Noise line preprocessing is considered in Section 3. Monte Carlo approach is applied and the results for different configurations and are shown in Section 4. Discussion related to the obtained results is provided in Section 5. The final conclusions are in Section 6.

2 The Viterbi Algorithm

The Viterbi algorithm is one of the dynamic programming algorithms [2]. It uses trellis with a set of nodes, and in this paper nodes are assigned to the pixels' grid. The computation of the trajectory (path) is based on the calculation of maximal value for paths allowed by Markov transition model [12]. Markov model is related to the trajectory shape, allowed by the trellis geometry and include probabilities of transition between nodes. In this paper it is assumed that the line is one direction starting from the bottom row of the image toward top row. Many applications requires estimation the most valuable bottom part of image – nearest to robot. Robots control is based on the this area, and top part of image could be significantly deteriorated due perspective camera view.

Image analysis starts from first row $n = 1$ and zero values V are assigned to the nodes:

$$V_{n=1}, . = 0, \tag{1}$$

where V_n, x is the largest value assigned to the node n, x and the coordinates correspond to image coordinates. The assumed trellis is shown in Fig. 1.

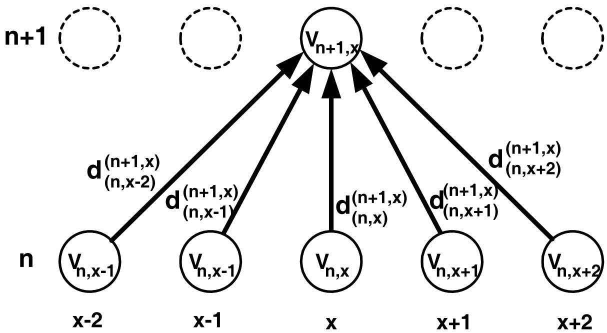

Fig. 1. Local paths in example trellis

In this paper local cost $d_{n,x+g}$ is related to the pixel value. The next row of values $n = 2$ is computed using the following formula, that uses local cost d:

$$V_{n+1,x} = \max \left(V_{n,x+g} + d_{n,x+g}^{n+1,x} \right). \tag{2}$$

for set of transitions:

$$g \in \{-2, -1, 0, +1, +2\} \tag{3}$$

The selection of the best local path to the particular node is preserved additionally:

$$L_n^{n+1,x} = \arg\max_g \left(V_{n,x+g} + d_{n,x+g}^{n+1,x} \right), \tag{4}$$

where L is local transition. The projection of values from first row is processed up to selected arbitrary n_{max} row. After reaching of this row the first phase (forward phase) of the Viterbi algorithm is achieved. The solution is the node with maximal value

$$P_{opt} = V_{n=n_{max},.}, \tag{5}$$

and the node number is:

$$x_{n=n+max} = \max_x \left(V_{n=n_{max},x} \right). \tag{6}$$

The second phase of the Viterbi algorithm (backward phase) is applied for the calculation of the best and first local transition. The obtained path of n_{max} length is rejected and only mentioned transition is valid.

The following recursive formula is applied for finding optimal solution:

$$x_{n-1} = x_n + L_{n-1}^{n,x} \tag{7}$$

for successively decremented row numbers:

$$n = n_{max}, \cdots, 2. \tag{8}$$

The transition between first and second row for x_1 point is the result of the Viterbi algorithm. The sliding window approach is applied in next step so overall process starts from next row with n_{max} depth analysis. The obtained set of points could be processed additionally using another algorithm for the trajectory filtering if it is desired.

3 Noise Line Preprocessing

The line is typically observed as a positive signal in the input image. Noise signal requires additional preprocessing and it could be obtained by the application of local filter that estimates noise parameter. As a local filter 2D local standard deviation $S(y, x)$ is applied:

$$S(y,x) = \sqrt{\frac{1}{(2x_N+1)(2y_N+1)} \sum_{\Delta y=-y_N}^{y_N} \sum_{\Delta x=-x_N}^{x_N} (\overline{X}_{y,x} - X_{y+\Delta y, x+\Delta x})^2}, \tag{9}$$

where X denote input image and $\overline{X}$ denote mean value for the corresponding area. The scaling coefficient $(2x_N+1)(2y_N+1)$ could be omitted for the efficient implementation. The window of the local estimation of standard deviation has $(2x_N + 1) \times (2y_N + 1)$ size. Overall system is shown in Fig. 2.

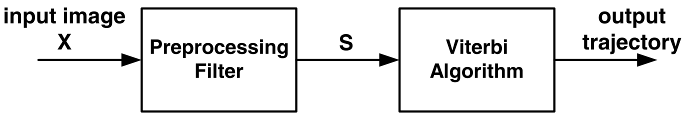

Fig. 2. Schematic of tracking system

4 Results

In Fig. 3 a single example tracking scenario is presented. The line is tracked from row 1 to 120, because the deep of analysis $n_{max} = 80$ is assumed. This line has standard deviation 1.0 and image is disturbed by a few lines with standard deviation selected by the random number generator from 0.2 to 1.0 range. All lines are disturbed by additive Gaussian noise and all lines are at the edge of visibility by human. The application of the filter that calculates local standard

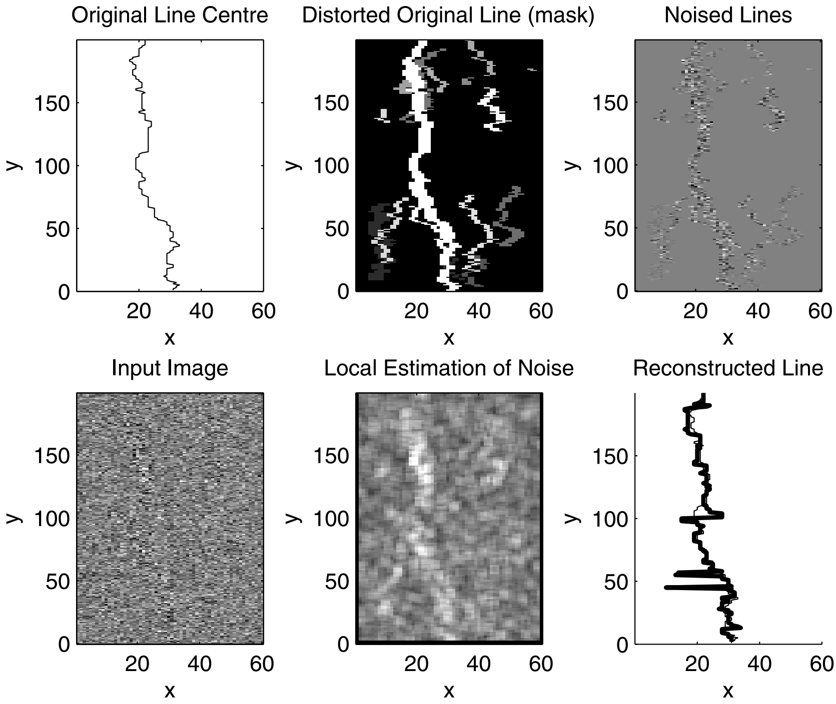

Fig. 3. Example tracking case

deviation allows the partial detection of the lines. The Viterbi algorithm gives the trajectory estimation that is comparable to the original.

The window size of the preprocessing filter is $H \times W = 5 \times 3$, where W and H are the width and height respectively.

Monte Carlo approach is applied for the analysis of the performance of proposed tracking system. This approach allows the testing of system before implementation for different scenarios.

In Fig. 4 the results for the different n_{max} values and for different noise values are shown.

5 Discussion

Four filters are compared (Fig. 4) using Monte Carlo approach and there are 300 cases in each configuration. Smooth line on figures means that test is sufficient. The window size of the filter influences the result. The filter with larger window height (11) gives better results. This is the result of the directional (vertical) type of line. The fitting of filter window to the direction of the line improves the detection. The width of filter window influences the results also. The width of line is variable, but it allocates a few pixel typically. Larger width of window reduces the tracking performance, because narrow windows are better fitted inside line.

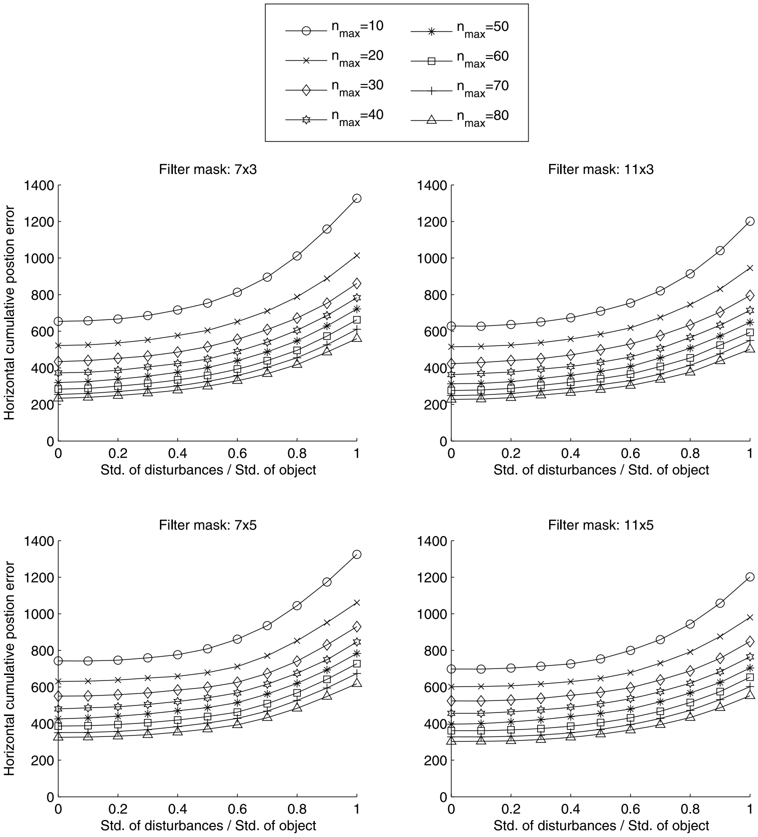

Fig. 4. Cumulative error for horizontal position for different filters and noise conditions

The influence of the deep of analysis is well visible (Fig. 4). Larger n_{max} values give better results, but the benefits of performance is reduced for $n_{max} > 30$. The selection of the proper n_{max} value depends on available computation power and the cost is linear.

6 Conclusions

The application of the Viterbi algorithm with preprocessing allows the noise line tracking. Assumed local filtration based on estimation of the standard deviation allows the conversion of image to the desired measurement space.

There are many TBD algorithms and similar techniques could be applied, e.g. Particle Filters algorithm [6].

Filtering may work also as TBD algorithm if is applied in the specific direction and it is a kind of the hierarchical preprocessing [11]. The application of additional processing between both algorithms is possible also for the emphasis of the lines, using techniques shown in [8,4] for example.

The variable line width could be processed by the application of the filter banks [13] for the selection of best filtering space, instead processing of image with fixed filtering mask.

The computation cost is low for both parts and image could be processed using modern processors or microcontroller. Large images could be processed using GPGPUs (General–Purpose Graphics Processor Units) in real–time [9,17]. There are the Viterbi algorithm accelerators that are available in specific DSPs (Digital Signal Processors). Parallel processing of both algorithms is possible also.

Acknowledgment. This work is supported by the UE EFRR ZPORR project Z/2.32/I/1.3.1/267 /05 "Szczecin University of Technology – Research and Education Center of Modern Multimedia Technologies" (Poland).

References

1. Astrand, B., Baerveldt, A.: A vision-based row-following system for agricultural field machinery. Mechatronics 15(2), 251–269 (2005)
2. Bertsekas, D.: Dynamic Programming and Optimal Control, vol. I. Athena Scientific (1995)
3. Blackman, S., Popoli, R.: Design and Analysis of Modern Tracking Systems. Artech House (1999)
4. Frejlichowski, D., Forczmański, P.: General shape analysis applied to stamps retrieval from scanned documents. In: Dicheva, D., Dochev, D. (eds.) AIMSA 2010. LNCS, vol. 6304, pp. 251–260. Springer, Heidelberg (2010)
5. Horan, B., Najdovski, Z., Black, T., Nahavandi, S., Crothers, P.: Oztug mobile robot for manufacturing transportation. In: IEEE International Conference on Systems, Man and Cybernetics (SMC 2011), pp. 3554–3560 (2011)
6. Huang, D., Xue, A., Guo, Y.: A particle filter track–before–detect algorithm for multi–radar system. Elektronika ir Elektrotechnika 19(5), 3–8 (2013)
7. Lech, P., Okarma, K.: Optimization of the fast image binarization method based on the monte carlo approach. Electronics and Electrical Engineering 20(4), 63–66 (2014)
8. Marchewka, A.: Crack detection on asphalt surface image using local minimum analysis. Advances in Intelligent and Soft Computing 84, 353–359 (2010)
9. Mazurek, P.: Optimization of bayesian track-before-detect algorithms for GPGPUs implementations. Electrical Review R 86(7), 187–189 (2010)
10. Mazurek, P.: Track-before-detect algorithm for noise objects. Measurement Automation and Monitoring 56(10), 1183–1185 (2010)
11. Mazurek, P.: Hierarchical track-before-detect algorithm for tracking of amplitude modulated signals. In: Choraś, R.S. (ed.) Image Processing and Communications Challenges 3. AISC, vol. 102, pp. 511–518. Springer, Heidelberg (2011)
12. Mazurek, P.: Code reordering using local random extraction and insertion (LREI) operator for GPGPU-based track-before-detect systems. Soft Computing 18(6), 1095–1106 (2013)
13. Mazurek, P.: Track–before–detect filter banks for noise object tracking. International Journal of Electronics and Telecommunications 59(4), 325–330 (2013)

14. Mazurek, P.: Preprocessing using maximal autocovariance for spatio-temporal track-before-detect algorithm. In: Choras, R.S. (ed.) Image Processing and Communications Challenges 5. AISC, vol. 233, pp. 45–54. Springer, Heidelberg (2014)
15. Okarma, K., Lech, P.: A fast image analysis for the line tracking robots. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 329–336. Springer, Heidelberg (2010)
16. Ollis, M.: Perception Algorithms for a Harvesting Robot. CMU-RI-TR-97-43, Carnegie Mellon University (1997)
17. Pietraszek, J., Szczotok, A., Kocyłowska, E.: Factorial approach to assessment of gpu computational efficiency in surogate models. Advanced Materials Research 874, 157–162 (2014)
18. Stone, L., Barlow, C., Corwin, T.: Bayesian Multiple Target Tracking. Artech House (1999)
19. Taubel, G., Yang, J.S.: A lane departure warning system based on the integration of the optical flow and Hough transform methods. In: 2013 10th IEEE International Conference on Control and Automation (ICCA), Hangzhou, China, June 12-14, pp. 1352–1357 (2013)
20. Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13(2), 260–269 (1967)
21. Zhang, J., Chambers, A., Maeta, S., Bergerman, M., Singh, S.: 3d perception for accurate row following: Methodology and results. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, November 3-7, pp. 5306–5313 (2013)